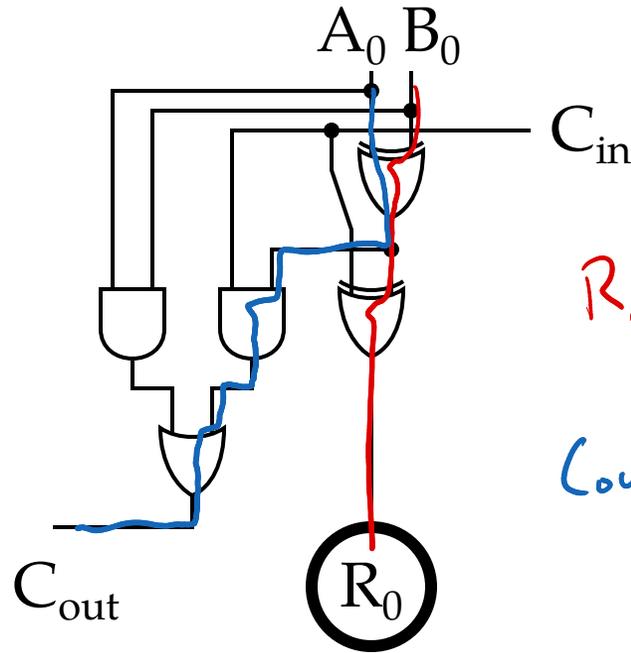


Warmup

What are the propagation delays for this circuit?



$$R_0 : 60 \text{ ps} + 60 \text{ ps} = 120 \text{ ps}$$

$$C_{out} : 60 \text{ ps} + 30 \text{ ps} + 40 \text{ ps} = 130 \text{ ps}$$

| Gate | t_{pd} (ps) | t_{cd} (ps) |
|--------------|---------------|---------------|
| NOT | 15 | 10 |
| 2-input NAND | 20 | 15 |
| 3-input NAND | 30 | 25 |
| 2-input NOR | 30 | 25 |
| 3-input NOR | 45 | 35 |
| 2-input AND | 30 | 25 |
| 3-input AND | 40 | 30 |
| 2-input OR | 40 | 30 |
| 3-input OR | 55 | 45 |
| 2-input XOR | <u>60</u> | 40 |

ES 4: Arithmetic circuits

Steven Bell

5 October 2023

Logistics stuff

Exam on Tuesday, study sheet on the website

You can bring one sheet of letter-size paper with your notes

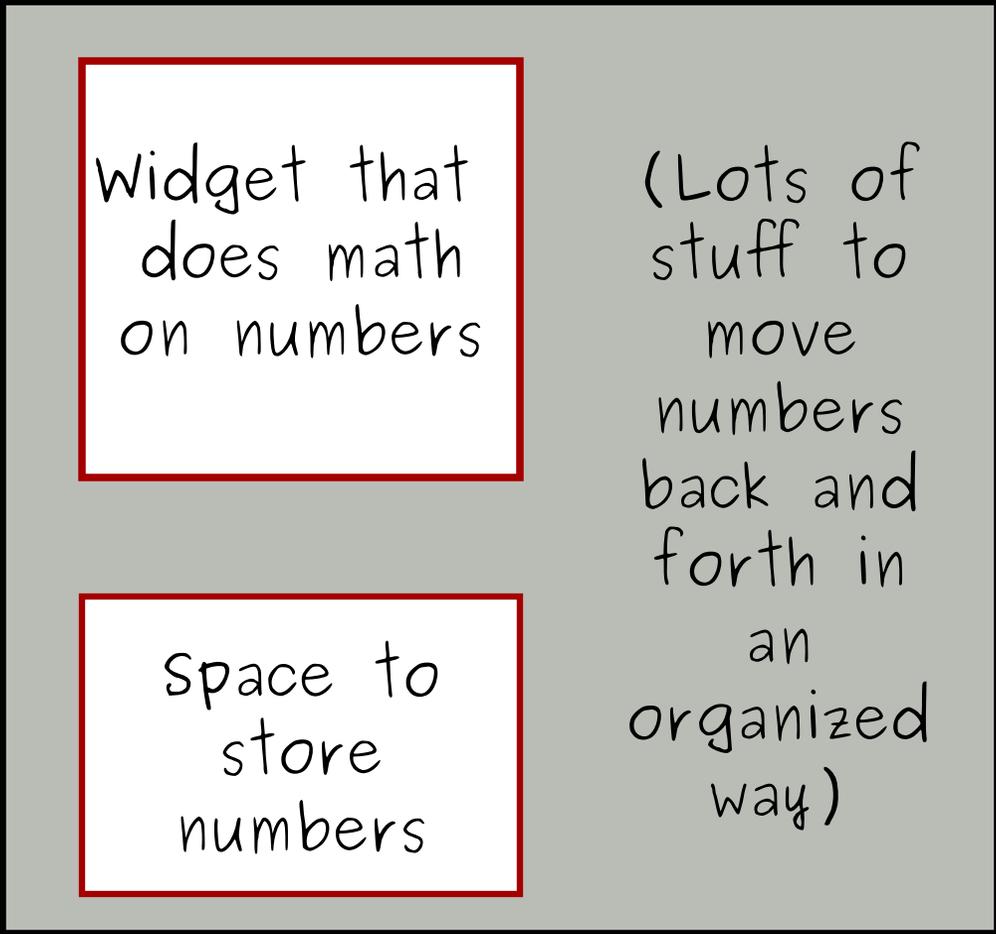
Watch Piazza for information on study sessions

Monday is a holiday

No labs, but watch Piazza for options

What's the point?

(over)simplified model of a processor



A diagram illustrating a simplified model of a processor. It consists of a large gray rectangle with a black border. Inside this rectangle, on the left side, are two smaller white rectangles with red borders. The top white rectangle contains the text "Widget that does math on numbers". The bottom white rectangle contains the text "Space to store numbers". To the right of these two white rectangles, within the gray area, is the text "(Lots of stuff to move numbers back and forth in an organized way)".

Widget that
does math
on numbers

Space to
store
numbers

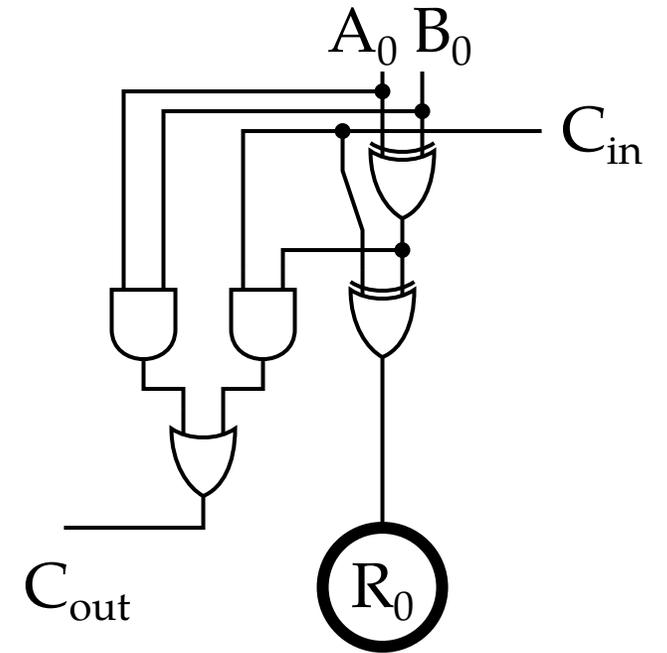
(Lots of
stuff to
move
numbers
back and
forth in
an
organized
way)

By the end of class today, you should be able to:

- Explain the basic principles of a carry-lookahead adder, and why it is faster than a ripple-carry adder.
- Explain what an ALU is, and describe the function of the control and flags signals.

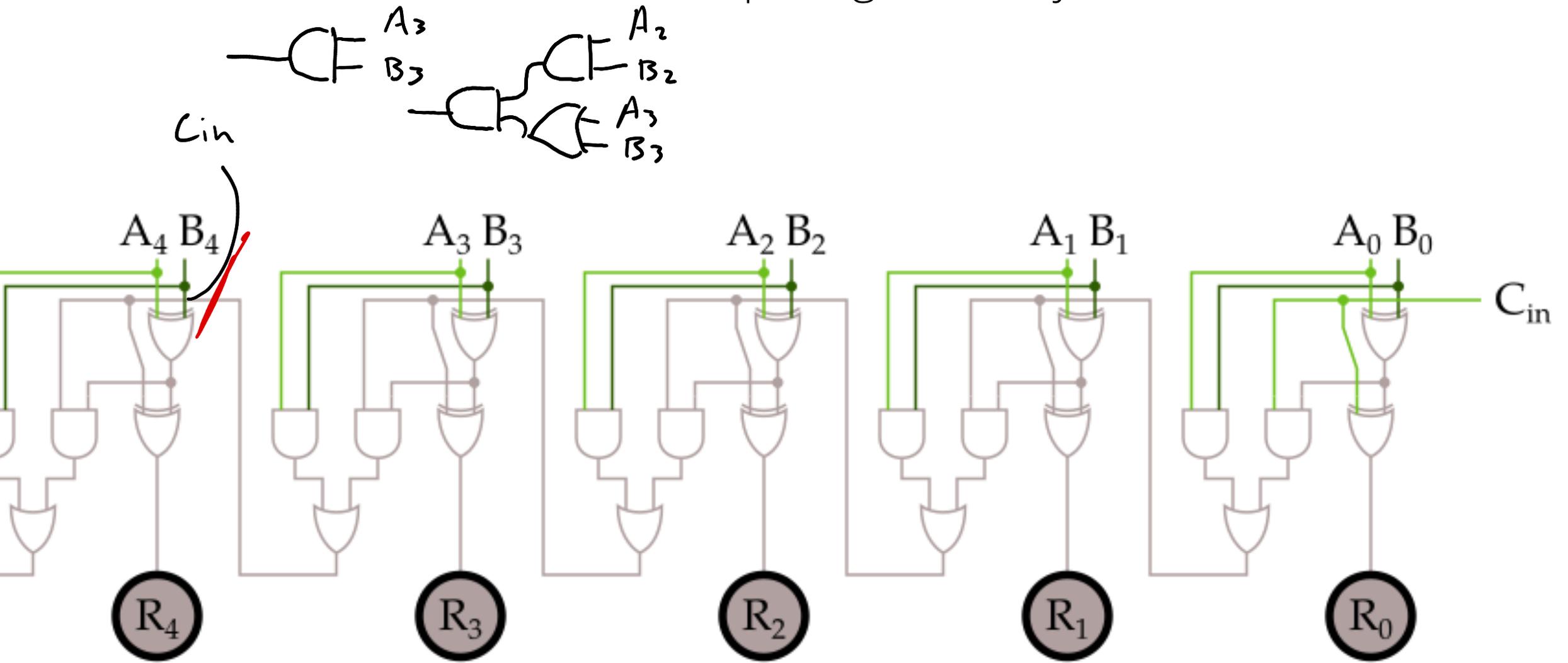
Ripple-carry adder

$$\text{result} = A \oplus B \oplus C_{in}$$

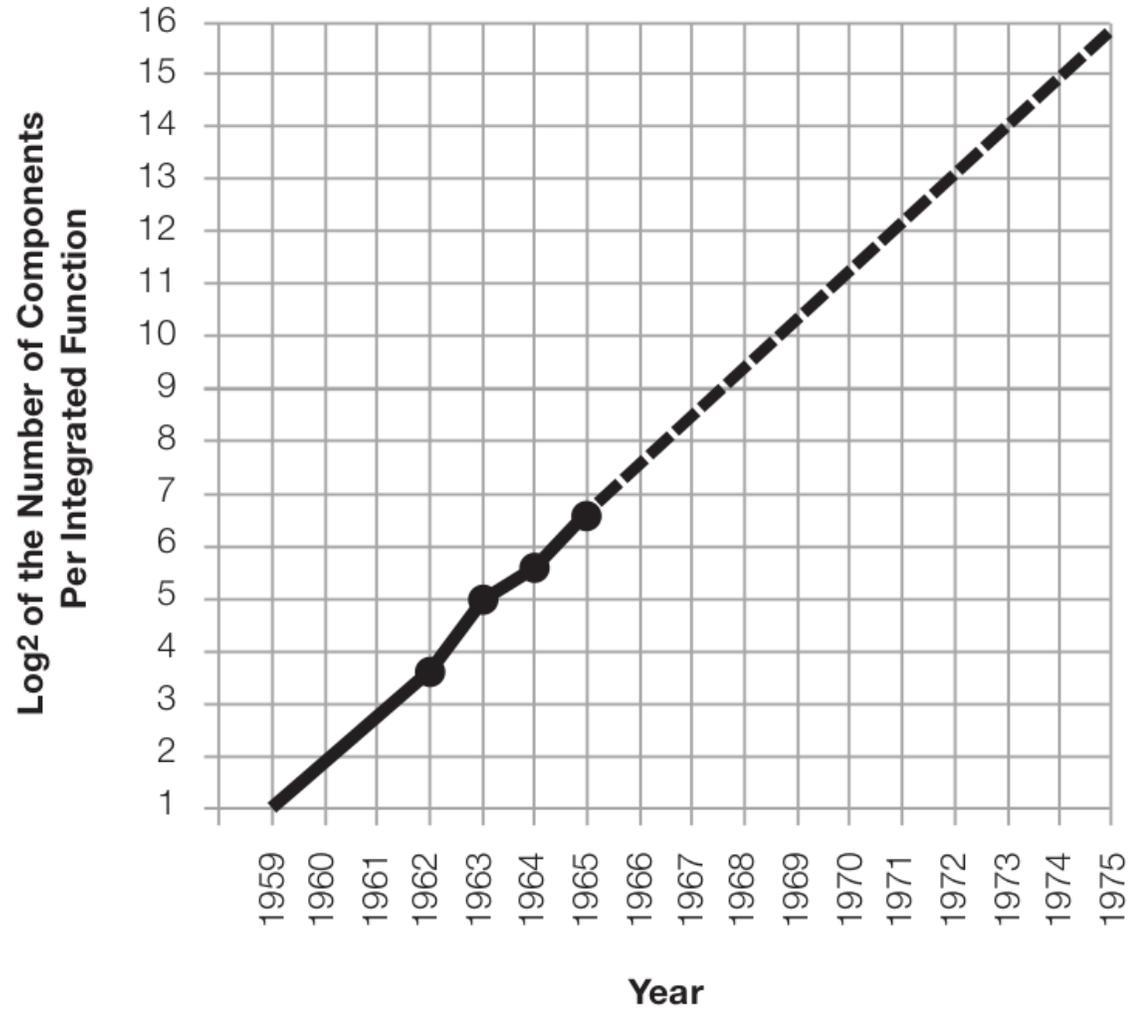


Carry-lookahead adder

What if we had a shortcut to computing the carry?



Moore's "law"



Gordon Moore, "Cramming more components onto integrated circuits" Electronics Magazine, April 1965

Prefix adder

int x[10]

x[3] = 2;

Multiplication

Works just like multiplication in decimal

$$\begin{array}{r} \text{A} \quad \begin{array}{r} 001 \\ 000 \\ 101 \end{array} \quad \begin{array}{r} \text{B} \\ 111 \\ 101 \end{array} \\ \hline \begin{array}{r} 001 \\ 000 \\ 101 \end{array} \times \begin{array}{r} 111 \\ 101 \end{array} \\ \hline \begin{array}{r} 001 \\ 000 \\ 101 \\ 000 \\ 000 \end{array} \\ \hline \begin{array}{r} 100011 \\ 32162421 \end{array} = 35 \end{array}$$

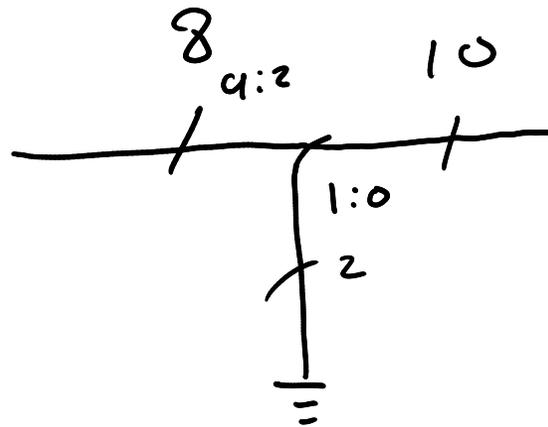
Multiplication, faster

Write out some numbers in binary and multiply them by 4

Design a circuit that multiplies numbers by 4

$$\begin{array}{r} 3 \quad 011 \\ \times 4 \quad 100 \\ \hline 000 \\ 000 \\ 011 \\ \hline 01100 = 12 \end{array}$$

00
always 00



5
+
a bus with 5 bits

Challenge: design a circuit that multiplies numbers by 17

Division

Is slow and uses lots of hardware

But dividing by certain values is really easy...

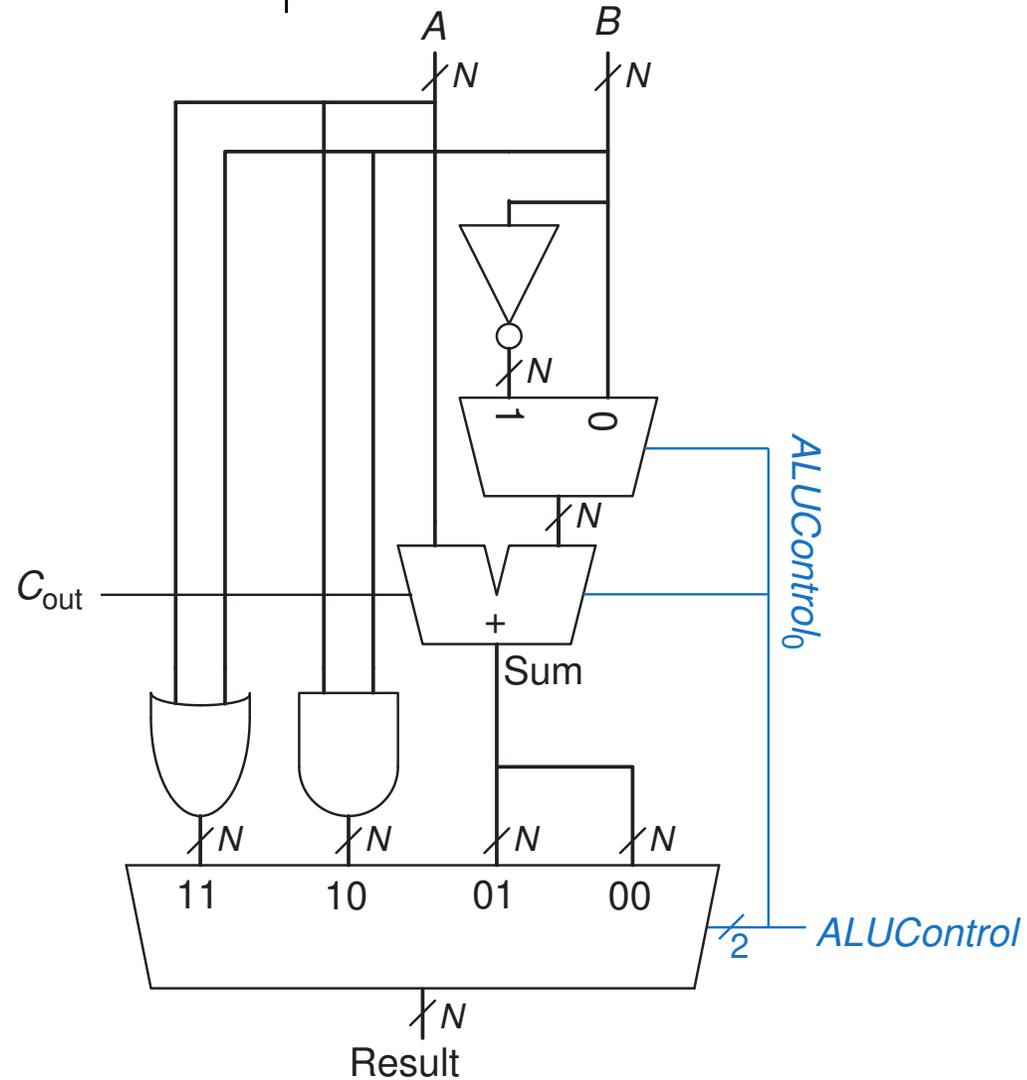
What about negative numbers?

Arithmetic Logic Unit (ALU)

Don't memorize this, it's just one example

ALUControl

- 0 0 Add
- 0 1 Subtract
- 1 0 AND
- 1 1 OR



Where are we?

We've built lots of stuff with combinational logic

We can even build the core of a computer

After the exam, we're going to talk about storage